

NAPREDNI OPERACIJSKI SUSTAVI

10.(OS glavna skripta) Komunikacija između procesa

- razlozi za međusobnu komunikaciju procesa: obavljanje zajedničkog posla, korištenje dijeljenih sredstava

10.1. Unutar istog računalnog sustava

- procesi imaju namjerno razdvojene adresne prostore, za razliku od dretvi unutar jednog procesa
- najjednostavnije rješenje pomoću **datoteka na disku** (jako statično)
- **dijeljeni spremnički prostor**
 - OS omogućuje dodjeljivanje dijeljenog prostora u dijelu virtualnog adresnog prostora više procesa
 - sve dretve iz oba procesa mogu pristupiti dijeljenom prostoru, ali ih je potrebno sinkronizirati
 - u slučaju fork-a, dijete nasljeđuje zajednički prostor
 - dva nepovezana procesa trebaju znati zajednički ključ koji određuje dijeljeni prostor
- **razmjena poruka između procesa**
 - poruke se ne šalju direktno drugom procesu, već se stavljaju u **red poruka**
 - `poslati_poruku(m)` i `prihvatiti_poruku(m)` (blokirajući poziv prihvatanja)
- **varijable okoline**
 - niz znakova oblika `ime=vrijednost` koji se predaje svakom programu prilikom pokretanja
 - različiti shellovi imaju drukčiju sintaksu rada, primjer za `sh`:
 - `ime=vrijednost` – postavlja varijablu lokalno (ne prosljeđuje ju pozvanim programima)
 - `export ime=vrijednost` – postavlja varijablu globalno (prosljeđuje ju pozvanim programima)

Cjevovodi

- dvosmjernan FIFO komunikacijski mehanizam unutar OS-a
- čitanje je destruktivna operacija, cjevovod može biti prazan ali i pun!

prespajanje na cjevovod

<code>close(w);</code>	<code>close(r);</code>
<code>close(stdin);</code>	<code>close(stdout);</code>
<code>dup(r);</code>	<code>dup(w);</code>
<code>close(r);</code>	<code>close(w);</code>

- **neimenovani (obični) cjevovodi**
 - procesi koji komuniciraju moraju biti u srodstvu (cjevovod se nasljeđuje fork-om)
 - 2 opisnika – jedan za čitanje, jedan za pisanje (zatvoriti onaj koji procesu nije potreban)
 - `stdin` ima opisnik 0, `stdout` ima opisnik 1, `stderr` ima opisnik 2, dalje stvoreni cjevovodi
 - to se može upotrijebiti za prespojiti `stdin/out/err` na cjevovod (vidi okvir iznad)
 - buffer veličine cluster (4 kB) koji se može napuniti
 - mogu se upotrijebiti na terminalu za povezivanje `stdout` i `stdin` dva procesa (npr. `ls | wc`)
- **imenovani cjevovodi**
 - `*.p` datoteka na disku koristi se kao cjevovod
 - postoji neovisno o procesima i tako omogućuje povezivanje procesa koji nisu u srodstvu
 - svaki proces ga otvara kao datoteku u načinu čitanja ili pisanja, time dobivajući vlastiti opisnik
 - maksimalna veličina određena maksimalnom količinom slobodnog prostora na disku

10.2. Raspodijeljeni sustavi

- procesi na različitim računalima žele međusobno komunicirati na mreži (razmjena poruka, podataka)
- problem: nema zajedničkog OS-a ni mogućnosti dijeljenja adresnog prostora

Osnove umrežavanja i struktura interneta

- protokoli – utvrđena pravila komuniciranja
- protokolni stog – skup protokola na različitim razinama potrebnih za funkcionalno povezivanje
 - primjenska razina (application layer) – protokol pojedine mrežne usluge (HTTP, FTP, SMTP...)
 - prijenosna razina (transport layer) – pravila prijenosa podataka
 - UDP – jednostavniji, koristi se kod manjih količina podataka
 - TCP – dodatno nudi pouzdan prijenos podataka, oporavci od pogreške, očuvanje poretka
 - mrežna razina (internet layer) – adrese izvoršnog i odredišnog računala (IP)
 - razina prijenosnih putova (link layer) – fizički prijenos podataka
- nekoliko kontinentalnih okosnica (backbones) sastavljenih od vrlo brzih komunikacijskih veza i routera
- okosnice su povezane međukontinentalnim vezama, a na njih su povezane nacionalne i regionalne mreže
- na nacionalne i regionalne veze su preko gatewaya spojene lokalne mreže koje se sastoje od računala
- svako računalo ima jedinstvenu IP adresu
- paketi od izvorišta do odredišta mogu pristizati različitim putovima i redoslijedima – TCP ih mora poredati

Načini komunikacije raspodijeljenih sustava

- **razmjena poruka** (asimetrično)
 - prvi proces (host) uspostavlja imenovanu komunikacijsku priključnicu (**socket**) i čeka na zahtjev
 - drugi proces (guest) pokušava uspostaviti vezu na isti socket
- **poziv udaljenih procedura** (RPC)
 - oponaša pozivanje potprograma pomoću već opisanog mehanizma razmjene poruka
 - klijent poziva proceduru s podacima, a poslužitelj izvršava proceduru i vraća rezultate klijentu
 - spojni modul (stub) oblikuje i proslijeđuje poruke i rezultate između njih putem komunikacijskog sustava
 - podaci se moraju prenositi vrijednošću (call by value) radi nemogućnosti dijeljenja prostora
- **raspodijeljeni dijeljeni spremnički prostor**
 - sinkronizacija sadržaja prostora putem razmjene poruka (objavljivanje promjena)
 - problem: ako proces piše u prostor, treba izvršiti promjene u svim kopijama kod drugih procesa
 - radi kašnjenja u komunikaciji, dolazi do pojave razlika (nekonzistentnosti)
 - zbog toga se još istražuje i nije u praktičnoj primjeni

10.3. Mutex kod raspodijeljenih sustava

Throwback: mutexi unutar jednog računala

- unutar jezgre:
 - jednoprosorski sustav: zabranom prekida
 - višeprosorski sustav: ispitivanje zastavice u dijeljenom prostoru za ulaz u jezgru, inače radno čekanje
 - dvije najvažnije jezgrine funkcije: ispitivanje binarnog semafora i postavljanje binarnog semafora
 - unutar programa/procesa: pozivanjem jezgrenih funkcija koje implementiraju binarni semafor
- u raspodijeljenom sustavu ne postoje te mogućnosti, mogu se samo razmjenjivati poruke

Očuvanje vremenskog uređenja (redoslijeda) događaja

- **lokalni logički sat** – u svakom čvoru postoji logički sat koji se povećava nakon svakog bitnog događaja
 - pomoću njega se jednoznačno utvrđuje redoslijed događaja unutar jednog procesa
 - događaj slanja poruke od jednog procesa uvijek je prije događaja primitka iste poruke u drugom procesu
- **globalni logički sat**
 - kada proces šalje poruku on uz nju prilaže vrijednost svog logičkog sata
 - kada drugi proces primi poruku on postavlja vrijednost svog logičkog sata na **$\max(\text{svoj} + 1, \text{primljeni})$**
 - ako dva lokalna sata imaju jednake vrijednosti i njihove poruke se mimoidu, tiebreaker je indeks procesa

Protokoli za ostvarivanje mutexa

- **centralizirani protokol**
 - jedan proces u mreži se je odgovoran za ostvarenje mutexa
 - ostali ga traže dozvolu za ulazak u K.O.
 - centralni proces će poslati odobrenje nekom procesu kada može ući u K.O.
 - kada taj proces izlazi iz K.O., šalje poruku o izlasku centralnom procesu
 - centralni čvor organizira red pristiglih zahtjeva po kojem dopušta ulazak u K.O.
 - nedostatak: velika ovisnost o centralnom procesu
- **putujuća značka**
 - definira se značka koja ciklički putuje kroz sve čvorove
 - kada proces primi značku, zadržat će je dok izvršava K.O. (ako želi), pa je onda prosljediti dalje
 - nedostaci:
 - ispad bilo kojeg procesa bi prekinuo proces prosljeđivanja
 - ako ni jedan proces ne želi u K.O., značka se stalno prosljeđuje i nepotrebno opterećuje mrežu
- **Lamportov raspodijeljeni protokol**
 - svaki proces održava vlastiti red zahtjeva za ulazak u K.O. koristeći globalni logički sat
 - kada proces želi ući u K.O., šalje **zahtjev** za ulazak svim ostalim procesima s vlastitim lokalnim satom
 - kada drugi proces primi zahtjev, stavlja zahtjev u svoj red i šalje **odgovor** s novim lokalnim satom
 - nakon primanja svih odgovora, prvi proces smije ući u K.O. kada je njegov zahtjev na početku reda
 - kada proces izlazi iz K.O., uklanja svoj originalni zahtjev iz svog reda i šalje **izlazak** ostalim procesima (kako bi i oni mogli ukloniti njegov zahtjev)
 - potrebno razmijeniti $3 \cdot (N - 1)$ poruka za jedan ulazak i izlazak iz K.O.
- **protokol Ricart-Agrawala** (nadogradnja Lamportovog)
 - kada proces želi ući u K.O., šalje **zahtjev** za ulazak svim ostalim procesima s vlastitim lokalnim satom
 - kada drugi proces primi zahtjev, usklađuje svoj lokalni sat
 - **odgovor** šalje ako ne želi ulaziti u K.O. ili ako je vlastiti zahtjev došao nakon primljenog (po satu)
 - kada proces primi neki odgovor, povećava svoj lokalni sat za 1
 - nakon primanja svih odgovora, prvi proces smije ući u K.O.
 - kada proces izlazi iz K.O., šalje odgovor svim procesima čiji zahtjevi od njega čekaju na odgovor
 - potrebno razmijeniti $2 \cdot (N - 1)$ poruka za jedan ulazak i izlazak iz K.O.